



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/802,239

03/17/2004

Alex A. Kipman

13768.1375

5256

47973

7590

06/18/2010

WORKMAN NYDEGGER/MICROSOFT

1000 EAGLE GATE TOWER

60 EAST SOUTH TEMPLE

SALT LAKE CITY, UT 84111

EXAMINER

WEI, ZHENG

ART UNIT

PAPER NUMBER

2192

MAIL DATE

DELIVERY MODE

06/18/2010

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/802,239	Applicant(s) KIPMAN ET AL.	
	Examiner ZHENG WEI	Art Unit 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 19 February 2010.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,3,5,6,8-16,18-20,22-26,28-30 and 37-45 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1,3,5,6,8-16,18-20,22-26,28-30 and 37-45 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

Detailed Action

Remarks

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 02/19/2010 has been entered.
2. This office action is in response to the amendment filed on 02/19/2010.
3. Claims 1, 3, 5-6, 8-16, 18-20, 23-26, 28-30 and 37-45 have been amended.
4. Claims 4 and 31-36 have been cancelled.
5. Claims 37-45 have been added.
6. Claims 1, 3, 5-6, 8-16, 18-20, 22-26, 28-30 and 37-45 remain pending and have been examined.

Response to Arguments

7. Applicant's arguments filed on 02/19/2010, in particular on pages 10-14, have been fully considered but they are not persuasive. For example:
 - At page 10, fourth paragraph – page 12, first paragraph, Applicants argue that the Ant tool is only a scripting tool for automating the build process, but is not the actual “build” process and the Ant tool is not equating the claimed “build process processor”. Because Applicants believe that the actual build is

performed by the JDK (Java Development Kit). However, Examiner respectfully disagrees. First of all, Examiner wants to discuss the definitions of terms: process, build process and build process processor. According to Microsoft's own dictionary (Microsoft computer dictionary, fifth edition, p.423), "process" is "A program or part of a program; a coherent sequence of steps undertaken by a program" and "processor" is central processing unit or microprocessor that is used to execute computer instructions. Accordingly, the "build process" is the process including sequence of steps to make the software build and "build process processor" is merely the computer hardware to perform the sequence of the build steps as specified by the build process. Thus, it is clear that the Ant tool as Examiner stated before which comprises commands (Copydir, Mkdir, Cvs, Javac...) that direct the computer to compile and generate software build step by step when executed by the computer (see for example, Cymerman, pages 1-2, command "Mkdir" for generating a directory, command "Cvs" for checking out source files; command "Javac" for compiling source files to generate target files; Further see example code at page 3 and related text). Therefore, the Ant tool/script including the specified commands/steps for making software build is considered as the "build process". For the "build process processor", it is computer processor which is required to execute the Ant tool/script to perform and realize the functions/steps for the build process as defined in the Ant. It can be seen that the build process as defined by the Ant tool/script is equivalent to the

Art Unit: 2192

steps/functions performed by the “build process processor” when executing the Ant tool/script. Secondly, the JDK (Java Development Kit) as a Java programming tool/package, it includes compiler (javac), loader(java), archiver(jar), debugger(jdb) and other components for developing Java application and thus It is not the “build process” as it does not provide or specify the sequence of steps to generate software build. Moreover, Ant tool/script is a well-known and widely used software build tool in computer art to make Java software build as either admitted by Applicants’ IDS or prior arts published as US patents or PGPubs. (now made pf record, US 20040123268 A1, Fig.1, Ant Build Script, discloses a build process to generate software build from step of Java source code inputs to step of compilation; US 2005/0015762 A1, paragraph [0044], discloses the “Ant is a known Java based build tool that is manufactured by The Apache Software Foundation”). Therefore, Examiner’s position is that the Ant tool/script which defines the sequence of steps for making software builds as disclosed by Cymerman is the “build process” as Applicants claimed.

- At page 12, first paragraph, applicants submit that Ant does not disclose anything similar to limit a certain level of access during the build process. Examiner’s position is that Cymerman’s ‘include/exclude” and “unless” statements as disclosed in page 6 are used to perform condition checks to decide if it should include or exclude the specified files that match the pattern from the compilation. That is to say, the Ant tool/script/build process does

perform checks before the compilation in build process and the checking result directly determines the compilation results (class files) in destination directory (see for example, p.4, "destdir="\${build.classes}") as it determine whether to include or exclude the specified source files/build entities in compilation (see for example, p.4, "srcdir="\${src.dir}"). It can be seen that the only difference is using the checking step to check different attributes/trust level of the source files/build entities according to user's requirement or configuration before the compilation, and thus, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use the checking step to check source files' different conditions, e.g. "include/exclude" and "unless" a property is set to "true", or source files/entities' trust levels) to generate different software builds as user required. As discussed above, it is obvious that all of these methods either in prior arts or current claims are merely applying different user's rules or restrictions on the general known build process for different purposes, e.g., one person can use source files/build entities' trust levels as the build rules/restrictions, another one might use source files /build entities' file size as checking rules to define that if size is greater than 100kbyte, build fails, or likes Cymerman's restriction condition to check "include/exclude" and "unless" a property of the source files/build entities is set to true, then generate different software builds. Therefore, the build process and build rule can be combined to implement in different ways according to the user or design requirement.

Art Unit: 2192

- At page 12, second paragraph, Applicants submit that it makes no sense for making build with a trusted permission level or semi-trusted permission level to be consider as using include or exclude statement in Ant tool/script. However, Examiner's position is that the trust levels as recited in the claims, especially in the amended version, explicitly points out that the trust level within the policy component "is accessed by the build process processor before building the project" (see for example, p.2, lines 6-7 of claim 1). It can be seen that trust level checking is just an additional step to apply rules or restrictions to the general build process/project to determine if it continues executing the build process or aborts the build process before executing the build process/project. Prior arts Graham (2005/0198628, Fig.3, item 360 rules data store, 370 constraint data store; Fig.5, steps 50-580 and related text), Vasilik (2003/0163799 A1, fig.4, step 401-414 and related text) or McIntyre (6,178,564 B1 Fig.2,, steps 56-90 and related text) (now made of record), all disclose similar methods to access and apply policy or rules to the build entities before the compilation step of the build process/project and generate different software builds according to different build policies or rules. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use Cymerman's Ant build tool/script/process with user defined build policy/rules to generate different software build according to application's design requirement. A new ground of rejection is applied as followings.

Claim Rejections - 35 USC § 112

8. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

9. Claim 12 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 12:

Claim 12 recites the limitation "the registry" in second line. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

11. Claims 1, 3, 5-6, 8-16, 18-20 and 22-30 and 37-45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cymerman (Michael Cymerman, Automate your build process using Java and Ant) in view of Jerger (US 6,321,334) and in view of Vasilik (Vasilik et al., US 2003/0163799 -- now made of record)

Claim 37:

Cymerman discloses a method which a build process (sequence of build steps defined in Ant tool) is executed in an integrated development environment, the method comprising:

- receiving a command to build a project that includes one or more build entities (see for example, p.4, example command “% ant –buildfile simple.xml” and related text);
- accessing one or more policies (include/exclude/unless and bsf.present in simple.xml) to determine each of the one or more build entities (see for example, p.6, example code and first paragraph, “You can use the include/exclude entities inside the javac task to include/exclude files matching the pattern in the name attribute from the compilation. From the above example, you want to include files contained in any directory ending in .java but the same time, you want to exclude files named Script.java unless a property bsf.present is set to true...You set the bsf.present property using the following task...”),
- determine according to the policies to include or exclude the build entities into the compilation of the build process (see for example, p.6, example code and first paragraph)
- executing the build process with the determined one or more build entities (see for example, p.4, second paragraph, “The Ant process invoked in the following line will run through the simple.xml file until the deploy command is reached”; also see p.6, example code and first paragraph, “You can use the

Art Unit: 2192

include/exclude entities inside the javac task to include/exclude files matching the pattern in the name attribute from the compilation. From the above example, you want to include files contained in any directory ending in .java but the same time, you want to exclude files named Script.java unless a property bsf.present is set to true”),

but Cymerman does not explicitly disclose a processor of a computer system to execute the build process and one or more policy files which are used to execute, determine and compile the build entities. However, Vasilik in the same analogous art of making software build discloses such computer processor (104), policy files (105) that the processor (104) invokes build environment (106), compiles source files (304) and builds application using the policy files (105) (see for example, Fig.1, item 104, request processor, item 105, “Rule set”, item 106, “Build Env”, item 107 “Application” and related text; See Fig.3, step 304, “Compile Source File(s) To Generate Application” and related text; Also see, steps 410-414 for selecting highest priority rule for build process/steps and related text). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to understand that Cymerman's build process as defined in Ant script has to be received and executed computer processor to perform and realize the sequence of build steps. The policy files/rule sets are also required the computer processor to access/apply to the build process or build entities (source files), e.g. selecting highest priority rule (403) to generate the software build (application) based on the user's requirement as

Art Unit: 2192

suggested by Vasilik (see for example, paragraph [0023], "builds missing target files using prioritized build rules within rule set 105...The rule priorities may be assigned by a developer...depending upon the functionality for the target file."). Cymerman and Vasilik discloses rule sets or policies (priority or true/false of a property) that can be accessed and applied the computer processor o the build entities and direct the build process to generate different target applications or builds based on the rule sets/policies. But neither of them explicitly discloses the rule sets/policies are levels of trust, wherein the levels of trust include:

(i) a trusted level that places no restrictions on the build process, (ii) a semi-trusted level that places restrictions on the build process, but still allows the build process to execute, and (iii) an untrusted level that causes the build process to abort;

However, Jerger discloses such similar security policy that is stored on a host computer, (see for example, Figure 8, items 812 Unsigned Permissions, 814 Trusted Signed Permissions, 816 Untrusted Signed Permissions and related text). wherein the one or more build entities are each associated with one or more levels of trust, such that at build time, a principal permission level under which the build process executes is determined by analyzing the levels of trust associated with each of the build entities, and lowest level of trust of all involved build entities dictates the principal permission level for execution of the build process (see for example, Fig.13A-C, step 1312, "Is class digitally signed?", step 1324 "Fail", step 1334, "Compare Requested permission set to granted

Art Unit: 2192

permission set”, step 1338, 1318 “Grant requested Permissions”, “Store any Granted Permissions with the Class”). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to define those different levels of trust for the build entities according to the user’s or design’s requirements. Such user defined rule sets/policies including the priority or levels of trust are also obvious for a person in the art to apply to Cymerman’s or Vasilik’s build process in order to generate user preferred software build by the rule selecting step (403). One would have been motivated to do so to generate build according to the requirement (see for example, paragraph [0023], “builds missing target files using prioritized build rules within rule set 105...The rule priorities may be assigned by a developer...depending upon the functionality for the target file.”).

Claim 38:

Cymerman discloses the method of claim 37, further comprising sending a message when the build process fails process (see for example, p.7, section Reporting enhancement, “If you wanted to extend Ant’s functionality to provide notification when certain steps in the build process are completed or are in progress...”; “BuildListener” and related text)

Claim 39:

Cymerman discloses the method of claim 37, further comprising receiving input from a developer that defines criteria to be included in the one or more policy files (see for example, p.6, example code and first paragraph, "You can use the include/exclude entities inside the javac task to include/exclude files matching the pattern in the name attribute from the compilation. From the above example, you want to include files contained in any directory ending in .java but the same time, you want to exclude files named Script.java unless a property bsf.present is set to true...You set the bsf.present property using the following task..."), but Cymerman does not explicitly disclose the policy related to how a level of trust is assigned to a build entity entities. However, as discussed in claim 37, the levels of trust is just another form of the rule sets/policy that is obvious for Cymerman or Vasilik to include in their build process as both have user input feature to define user specified rule set or policy as discussed above.

Claim 40

Cymerman discloses the method of claim 39, wherein the criteria includes a location where a build entity is stored (see for example, p.6, lines 4-6, ".**/" location of files as in the form of path or location and related text).

Claims 41-42:

Cymerman, Vasilik and Jerger disclose the method of claim 37, but none of them explicitly discloses determining that the one or more policy files does not contain

Art Unit: 2192

criteria that define a level of trust for one of the build entities in the project, and assigning an untrusted level of trust to the build entity; wherein one of the one or more build entities is associated with at least two levels of trust, However, such information is merely the user's requirements, as Cymerman, Vasilik and Jerger disclose the method to define and apply user specified rule sets/policies, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use the above method to define different rule set/policies as user or application required.

Claim 43:

Cymerman, Vasilik and Jerger disclose the method of claim 37, Jerger further discloses wherein the one or more policy files includes a default policy file and a user-defined policy file (col.3, lines 55-65, "The user may utilize one or more predefined security zones, configure custom security zones, or do nothing and accept a default set of predefined security zones")

Claim 44:

Jerger further discloses the method of claim 43, wherein the user-defined policy file overrides the default file when a conflict occurs (col.3, lines 66 – col.4. line 12, "predefined zone security policy" and "default security zones policy").

Claim 45:

Art Unit: 2192

Cymerman, Vasilik and Jerger disclose method of claim 37, Jerger further discloses wherein at least one of the one or more policy files is stored with access restrictions (see for example, Fig.3, step 320, “store security configuration data for security zone in the system registry” and related text).

Claims 1, 3, 5-6 and 8-10:

Claims 1, 3, 5-6 and 8-10 are system version for performing the claimed method as in claims 37-45 addressed above, wherein all claimed limitation functions have been addressed and/or set forth above and certainly a computer system would need to run and/or practice such function steps disclosed by reference above, For example, Cymerman discloses in “What do I need to use Ant?” that it must install three components on the system (machine) to run Ant (p.2). Thus, they also would have been obvious.

Claims 11-19:

Claims 11-19 are another system version for performing the claimed method as in claims 37-45 addressed above, wherein all claimed limitation functions have been addressed and/or set forth above and certainly a computer system would need to run and/or practice such function steps disclosed by reference above. Thus, they also would have been obvious.

Claims 20 and 22-30:

Art Unit: 2192

Claims 20 and 22-30 are computer program products version of the claimed method, wherein all claimed limitation functions have been addressed in claims 37-45 above respectively. It is well known in the computer art that such method steps can be implemented as computer program and can be practiced and /or stored on a computer storage medium. Thus, they also would have been obvious in view of reference teachings above.

Conclusion

12. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.
13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Zheng Wei whose telephone number is (571) 270-1059 and Fax number is (571) 270-2059. The examiner can normally be reached on Monday-Thursday 8:00-15:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571- 272-1000.

Art Unit: 2192

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Z. W./
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192